The Thesis committee for Christopher Carroll Johnson Certifies that this is the approved version of the following thesis

Greedy Structure Learning of Markov Random Fields

APPROVED BY

SUPERVISING COMMITTEE:

Pradeep Ravikumar, Supervisor

Inderjit Dhillon

Greedy Structure Learning of Markov Random Fields

by

Christopher Carroll Johnson, B.S.C.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2011

Acknowledgments

I would like to thank my advisor, Pradeep Ravikumar, for inspiration, guidance, and encouragement on this work. In addition, I would like to thank Ali Jalali for his collaboration and work on the proof techniques and theoretical analysis used in this paper. Also, I would also like to thank Inderjit Dhillon and the students of his lab for motivation and many stimulating conversations regarding Machine Learning, Data Mining, and Statistics. Finally, I would like to thank my friends and family for their faith and encouragement in my many late nights of research and writing. I couldn't have finished this work without their support.

Greedy Structure Learning of Markov Random Fields

Christopher Carroll Johnson, M.S.C.S.
The University of Texas at Austin, 2011

Supervisor: Pradeep Ravikumar

Probabilistic graphical models are used in a variety of domains to capture and represent general dependencies in joint probability distributions. In this document we examine the problem of learning the structure of an undirected graphical model, also called a Markov Random Field (MRF), given a set of independent and identically distributed (i.i.d.) samples. Specifically, we introduce an adaptive forward-backward greedy algorithm for learning the structure of a discrete, pairwise MRF given a high dimensional set of i.i.d. samples. The algorithm works by greedily estimating the neighborhood of each node independently through a series of forward and backward steps. By imposing a restricted strong convexity condition on the structure of the learned graph we show that the structure can be fully learned with high probability given $n = \Omega(d \log(p))$ samples where d is the dimension of the graph and p is the number of nodes. This is a significant improvement over existing convex-optimization based algorithms that require a sample complexity of $n = \Omega(d^2 \log(p))$ and a stronger irrepresentability condition. We further

support these claims with an empirical comparison of the greedy algorithm to node-wise ℓ_1-regularized logistic regression as well as provide a real data analysis of the greedy algorithm using the Audioscrobbler music listener dataset. The results of this document provide an additional representation of work submitted by A. Jalali, C. Johnson, and P. Ravikumar to NIPS 2011 [10].

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 A Motivating Example

Suppose that we were charged with the task of recommending new music to a user based on their current music listening habits. If the user provided us with a sample of music that they enjoyed then how might we go about predicting new music that the user may also be interested in? This is a non trivial task that companies like Pandora Radio [19] and last.fm [13] have built their business model around. Suppose that we had access to a large corpus of data consisting of the listening habits of several million people. It's natural to assume that there are patterns in the listening habits of these users and that if we discovered these patterns then we could exploit them to recommend relevant new music. Figure 1.1 provides an example of possible music preference dependencies.

Listens to:	May also like:			
Led Zeppelin	The Doors	The Who	Jimi Hendrix	Cream
New Order	The Smiths	Joy Division	Jim Morrissey	The Cure
Fleet Foxes	Bon Iver	The Shins	Neutral Milk Hotel	Iron and Wine
Radiohead	Flaming Lips	The Smiths	Blur	The Pixies

Figure 1.1: Sample music preferences

If we were given a set of music preferences for a user, we may suggest new music to the user based on dependencies discovered in the data. The problem then becomes learning these dependencies given a large set of sample data.

This example provides us with a practical motivation for structure learning of Probabilistic Graphical Models, a topic which will be investigated in this paper. First, we will provide the reader with a preliminary outline of the Mathematical framework necessary to understand the depth of this work. We will then provide a brief outline of probabilistic graphical models including both undirected and directed graphical models. Next, we will present the structure learning problem and examine existing methods. We will then introduce a new, novel approach to learning the structure of a discrete, pairwise Markov Random Field using a greedy neighborhood selection algorithm. We will also provide an experimental analysis of the greedy method in comparison to another popular structure learning algorithm, ℓ_1-regularized logistic regression. Finally, we will return to our music recommendation example and attempt to fit a binary, pairwise Markov Random Field to the Audioscrobbler music listener dataset [9].

Before we can begin our discussion of Probabilistic Graphical Models and structure learning we must first lay down a foundation of background material regarding key concepts from probability theory. In particular, we will need to outline the concepts of *random variables*, *probability distributions*, and *conditional probability*.

1.2 Random Variables

Probability theory deals with the analysis of the likelihood of random *events* defined over a *sample space* Ω. For example, consider rolling a single fair die. The sample space of a single roll can be defined as $\Omega = \{1, 2, 3, 4, 5, 6\}$ and the possible random events are subsets of Ω. For example, we can consider the event that a 4 is rolled as well as the event that an even value is rolled. If an event is a single element of the sample space $\alpha \in \Omega$ then we refer to the event as an *atomic* event. The event that a 4 is rolled is an example of an atomic event.

When dealing with random events we are often interested in numerical descriptions of the events and not just the probability of the event occurring. In order to handle this we can assign numerical descriptions of events to what are referred to as *random variables*. For example, suppose that we roll 2 fair dice. Then, we can define a random variable X to take on the numerical result of the sum of the dice. Definition 1 provides a more formal definition.

Definition 1. *A random variable X defined over a sample space Ω is a function $X : \Omega \to \mathbb{R}$ that maps an event ($x \subseteq \Omega$) to a real value.*

In order to avoid any confusion, in this paper we will denote random variables by upper case letters X and the possible values that they take on by lower case letters x. Random variables that only take on values from a countable set (such as the integers) are referred to as *discrete* random variables. The random variable defined as the sum of the roll of 2 fair dice is an example

of a discrete random variable. Random variables that take on values from an uncountable set (such as the reals) are referred to as continuous random variables. As an example of a continuous random variable consider choosing a random US citizen and measuring their height exactly. This measured value can be any real value in a specific range and so represents a continuous random variable.

1.3 Probability Distributions

In analyzing random variables we are often interested in the probability (or belief) that a random variable takes on certain values. To formally describe the different probabilities of a random variable taking on various values we define a *probability distribution*.

Definition 2. *A probability distribution P defined on random variable X over a sample space Ω is a mapping from events $\alpha \subseteq \Omega$ to real values on the interval $[0, 1]$ such that $P(\alpha) \geq 0$ and $P(\Omega) = 1$.*

In our example of rolling a single fair die we had equal probability of rolling each of the atomic events in our event space $\Omega = \{1, 2, 3, 4, 5, 6\}$. Then, the probability distribution defined by the random variable representing the value of a single roll of a fair die can be defined as $\forall i \in \Omega,\ \mathbb{P}(X = i) = \frac{1}{6}$. When each possible atomic event has equal probability we call this the *uniform distribution*.

1.3.1 Discrete Probability Distributions

Probability distributions defined on discrete random variables can be understood by a *probability mass function (pmf)*.

Definition 3. *A probability mass function (pmf) $f_X(x)$ of a discrete random variable X is given by $f_X(x) = \mathbb{P}(X = x)$ for all $x \in \Omega$.*

A pmf assigns a probability belief to each possible atomic event in the event space such that the sum of those beliefs adds up to 1. In our dice example, the pmf was defined as $\forall i \in \Omega,\ \mathbb{P}(X = i) = \frac{1}{6}$.

1.3.2 Continuous Probability Distributions

Due to the fact that continuous random variables can take on an infinite number of values it wouldn't make sense to describe continuous random variables using a probability mass function since the individual probabilities of specific events would likely approach 0. Instead, probability distributions defined on continuous random variables can be understood by a *probability density function (pdf)*. In order to define a pdf we also need the notion of a cumulative distribution function (cdf).

Definition 4. *A cumulative distribution function (cdf) $F_X(x)$ of a continuous random variable X is given by $F_X(x) = \mathbb{P}(X \leq x)$.*

In other words, the cdf of a continuous random variable X defines the probability that X will take on a value less than or equal to a specific value.

Given the above definition for cdf we are now ready to define the probability density function (pdf) of a continuous random variable.

Definition 5. *A probability density function (pdf) $f_X(x)$ of a continuous random variable X is a function that satisfies $F_X(x) = \int_{-\infty}^{x} f_X(u)du$.*

The pdf essentially describes the probability that a continuous random variable will take on values in a specified range. For example, given random variable X and its pdf $f_X(x)$ we can define the probability that X takes on a value in the range from a to b by $\mathbb{P}(a \leq X \leq b) = \int_a^b f(x)dx$.

1.4 Joint Distributions

We have already outlined a framework for representing probability distributions on individual random variables, but suppose now that we were interested in describing a probability distribution over multiple random variables. As a concrete example, suppose that we are interested in classifying college students by several different features such as age, height, weight, gender, and ethnicity. Each of these features can be thought of as a random variable. Each random variable can then be defined by its own probability distribution. For example, the number of male and female college students may be roughly the same and so $\mathbb{P}(gender = male) = \mathbb{P}(gender = female) = \frac{1}{2}$. In addition to describing the distribution of each individual feature we can also describe the *joint distribution* of the set of features.

Definition 6. *A joint distribution over a set $X = \{X_1, ..., X_n\}$ of random*

variables is denoted by $P(X_1, .., X_n)$ and is the distribution that assigns probabilities to events that are specified in terms of these random variables.

When dealing with joint distributions we will often denote a vector of random variables using a bold faced letter such as $\mathbf{X}$. Then, $P(\mathbf{X})$ refers to the joint distribution over the vector $\mathbf{X}$. The joint probability distribution defines the probability that each random variable in the distribution takes on certain values. For example, we can consider the joint probability that a randomly chosen college student is 21 years old, 5'10", weighs 160 lbs, and is male. In representing the full joint distribution over a set of n discrete random variables we can define an n-dimensional tensor that represents all combinations of events over those random variables. Even in the simplest case in which each random variable is binary valued, the size of such a table would of course be exponential $O(2^n)$ in the number of random variables. It then begs the question as to whether or not there are more compact parameterizations of joint distributions. The simple answer is "yes" in many cases as we shall see.

1.4.1 Marginal Distributions

Given a joint distribution over a set of random variables $\{X_1, ..., X_p\}$ it is often desirable to consider the distribution of a single random variable X_i. In other words, we may be interested in the probability that a single RV X_i takes on certain values, regardless of the other variables. We call such a distribution a *marginal* distribution and it can be calculated by $P(X_i) = \sum P(X_i, X_j)$

summing over all possible values of all possible j. The process of calculating a marginal from a joint distribution is often referred to as *marginalization*.

1.5 Conditional Probability

To begin with a concrete example, consider choosing a random college student as outlined previously. Suppose again that we know *a priori* that the number of male and female college students are roughly equivalent. Let X be a random variable representing the gender of our randomly chosen student. Then, we have that $\mathbb{P}(X = male) = \frac{1}{2}$. Now, suppose we are given additional information that the student is 6'0" tall. Given, this information it is more likely that our randomly chosen student is male than female. In order to represent this intuition in probability theory we use *conditional probability*.

Definition 7. *The conditional probability of an event A given an event B, denoted by $\mathbb{P}(A \mid B)$, is the probability of event A occurring given that event B occurred. In terms of random variables the conditional probability that random variable X takes on value x given that random variable Y takes on value y, denoted by $\mathbb{P}(X = x \mid Y = y)$, is the probability of X taking on value x given that Y took on the value y.*

In considering conditional probability we have both a *prior* probability of an event as well as a *posterior* probability of an event given additional information. In our previous example it turned out that the conditional probability (posterior probability) of our student being male given that they were

6'0" tall was greater than the probability that the student was male without any additional information (prior), however this is not always the case. As we shall see when we discuss *independence* it is often the case that additional information does not affect probability.

1.5.1 Chain Rule and Bayes Rule

Given a joint distribution over 2 random variables $P(X, Y)$ we can represent this distribution using conditional probabilities by the chain rule.

Definition 8. *Given random variables X and Y and a joint distribution over them $P(X, Y)$, the chain rule states that $P(X, Y) = P(X)P(Y \mid X)$.*

It should be apparent that this definition is symmetric. For example, let X be a random variable denoting the gender of our student and Y be a random variable denoting the age of the student. Then, in defining the joint distribution we have that $P(X, Y) = P(X)P(Y \mid X) = P(Y)P(X \mid Y)$. We can further extend the chain rule to multiple variables by the following definition.

Definition 9. *Given a joint distribution over n variables $P(X_1, ..., X_n)$ the chain rule states that $P(X_1, ..., X_n) = P(X_1)P(X_2 \mid X_1)...P(X_k \mid X_1, ..., X_{n-1})$.*

By rearranging our definition of the chain rule we can define the conditional probability of a random variable X given Y as $\mathbb{P}(X \mid Y) = \dfrac{\mathbb{P}(X \cap Y)}{\mathbb{P}(Y)}$. This is sometimes referred to as the definition of conditional probability. From this we immediately arrive at *Bayes Rule*.

Definition 10. $\mathbb{P}(X \mid Y) = \dfrac{\mathbb{P}(Y \mid X)\mathbb{P}(X)}{\mathbb{P}(Y)}$

Bayes Rule plays a central role in probability theory as it provides a translation between $\mathbb{P}(X \mid Y)$ and $\mathbb{P}(Y \mid X)$.

1.5.2 Independence

Going back to our example of choosing a random student, suppose that we are given that the student is 21 years old. Then, consider the conditional probability that the student is male given that the student is 21 years old $\mathbb{P}(X = male \mid Y = 21)$. If we assume that age does not affect the likelihood of the student's gender being male or female (which is a likely assumption) then it turns out that $\mathbb{P}(X = male \mid Y = 21) = \mathbb{P}(X = male) = \frac{1}{2}$. When this is the case we say that the event of the student being male is independent of the event of the student being 21 years old. In terms of random variables we can also say that the random variable denoting the gender of the student is independent of the random variable denoting the age of the student.

Definition 11. *Two events A and B are independent, denoted by $A \perp B$ if and only if $\mathbb{P}(A \cap B) = \mathbb{P}(A) \cdot \mathbb{P}(B)$.*

In other words, two random variables A and B are independent if the probability of one taking on specific values has no correlation with the other taking on specific values. It should be apparent that independence between 2 variables is symmetric. That is, $A \perp B$ if and only if $B \perp A$.

1.5.3 Conditional Independence

Consider a joint distribution over multiple random variables. Given 2 variables from our distribution it is often the case that they are not directly independent of each other but that they are *conditionally independent* given one or more other variables from the distribution. For example, consider the random variables A, S, and C where $A \in \mathbb{N}$ denotes the age of a person, $S \in \mathbb{N}$ denotes the number of years the person has been a smoker and $C \in \{0, 1\}$ denotes whether or not the person has lung cancer. The variables A and C and not necessarily independent since older people are more likely to have lung cancer (they've had more time to smoke and develop lung cancer). However, suppose that we were given the value of S. Then, in determining the probability that a person has cancer it probably wouldn't matter what their age was. In other words $\mathbb{P}(C \mid A, S) = \mathbb{P}(C \mid S)$. In this case we would say that C is conditionally independent of A given S, denote by $(C \perp A \mid S)$.

1.5.3.1 Pairwise vs Higher Order Dependencies

When being given the value of one random variable has an affect on the probability that another random variable takes on a specific value then we say that the variables are *dependent*. This is to say that the value a variable takes on is *dependent* on the value the other takes on. When the dependence is between 2 variables we call it *pairwise dependence*. However, dependence is not limited to 2 variables. We can also consider higher order dependencies consisting of multiple variables. In fact, we can even consider higher order

dependencies where none of the variables are pairwise dependent (every pair of variables are *pairwise independent*). Consider the variables X, Y, and Z such that $X \in \{0,1\}$, $Y \in \{0,1\}$, and $Z = X \oplus Y$ is the parity between X and Y. Suppose that both X and Y are uniformly distributed. Then, given that say $Y = 0$, the probability that $X = 0$ is still simply $\frac{1}{2}$. In other words, X and Y are pairwise independent. Also consider the pairs X and Z or Y and Z. Each pair is symmetric so lets focus on the pair X and Z. Again, being given the value of Z tells us nothing about the value of X. X is still uniform over the set $\{0,1\}$ because we weren't given the value of Y which has a uniform chance of being either 0 or 1. Then, for each pair of variables we have pairwise independence. However, these variables are not 3-way independent. If we were given any 2 of them, say X and Y, then the third variable is completely dependent on the values of the other 2. Then, we have a set of 3 variables such that each pair is pairwise independent but together there is a 3-way dependence among them.

Chapter 2

Probabilistic Graphical Models

In the previous chapter we discussed random variables and joint probability distributions defined over sets of random variables. We also mentioned that given a joint distribution over a set of p random variables $X_1, ..., X_p$, that in naively representing the joint distribution by listing every combination of settings of the variables, that such a p-dimensional tensor would be exponential $O(c^n)$ in the number of variables in the distribution. For example, even in the case when each random variable $X_i \in \{0, 1\}$ can only take on 2 values, naively specifying the joint distribution requires $2^n - 1$ values (the probabilities of each of the 2^n different assignments of values to the variables $X_1, ..., X_n$ minus 1 which can be calculated from the others). In all practical applications, storing an exponential amount of data quickly becomes unmanageable both to store and to work with, such as computing marginals or other inferences from the model. Additionally, learning each probability in the distribution from samples requires an enormous amount of data to accurately learn each parameter. These issues provide motivation for having more compact representation models and methods of inference for large multivariate distributions. Probabilistic Graphical Models provide a framework both for compact representation as well as efficient methods of inference on joint

distributions. They achieve this by taking advantage of independence and dependence relationships in distributions such as those that we looked at in the previous chapter. Generally, graphical models represent a joint distribution using vertices to represent variables in the distribution and edges to represent dependencies among the variables. In this paper we will focus on a specific type of undirected graphical models called *Markov Random Fields* though we will first give an outline of directed graphical models, also known as *Bayesian Networks*, as well as another type of undirected graphical models called *Factor Graphs*.

2.1 Bayesian Networks

As we have already shown in our discussion of the chain rule a joint probability distribution $P(\mathbf{X})$ can be factorized as a product of terms where each term is the probability of one variable, given all those *before* it in some order.

$$P(\mathbf{X}) = P(X_1)P(X_2 \mid X_1)...P(X_n \mid X_1, X_2, ..., Xn) \tag{2.1}$$

Also recall that by the definition of conditional independence that if $(X_i \perp \mathbf{X}_{\setminus X_j} \mid X_j)$, where $\mathbf{X}_{\setminus X_j}$ is syntactic sugar used to denote $\mathbf{X}$ *without* X_j, then $P(X_i \mid \mathbf{X}_{\setminus X_j}) = P(X_i \mid X_j)$. Bayesian networks aim to represent a joint distribution using these notions by factorizing the distribution by its dependencies. A Bayesian Network $\mathcal{B} = \langle G, P \rangle$ can be defined as a tuple

consisting of a directed acyclic graph (DAG) G and a set of conditional probabilistic distributions $P = P(X_1 \mid X_{\pi(1)}), ..., P(X_n \mid X_{\pi(n)})$ where $X_{\pi(i)}$ denotes the set of "parents" of variable X_i [18]. The parents of a variable X_i are precisely those variables such that X_i is conditionally independent of all other variables in the distribution that come *before* it in some ordering given its set of parents. In other words, we can define $X_{\pi(i)}$ by:

$$P(X_i \mid X_1, X_2, ..., X_{i-1}) = P(X_i \mid X_{\pi(i)}) \tag{2.2}$$

Then, applying the definition of conditional independence we get that a Bayesian Network factorizes as:

$$P(\mathbf{X}) = \prod_i P(X_i \mid X_{\pi(i)}) \tag{2.3}$$

It is important to note that 2.2 does not make any reference to the variables $X_{i+1}, .., X_n$ that come after it in our imposed ordering. We denote the full set of variables such that X_i is independent of the rest of the distribution given this set as the "markov blanket" of X_i. As it turns out, the markov blanket of a variable X_i consists of its parents, its children, and its children's parents [18].

The graph structure of a Bayesian Network $G = (V, E)$ can be defined by a DAG as follows. The set of vertices V is precisely the set of variables in

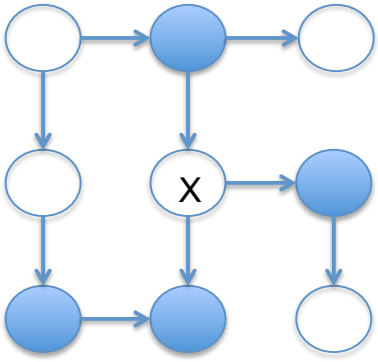

Fig 2.1: Graphical representation of a Bayesian Network defined over a joint probability distribution. The "markov blanket" for the variable represented by the X consists of the shaded vertices.

the distribution, $\{X_1, ..., X_n\}$. The set of edges E consists of a set of directed edges $\bigcup_i \{(X_j, X_i) : j \in X_{\pi(i)}\}$ from each vertex to each of its children.

Figure 2.1 provides a visual representation of a Bayesian Network along with the markov blanket of a single variable.

2.2 Factor Graphs

A Factor Graph is a type of undirected graphical model (edges are undirected as opposed to directed) that designates vertices for both random variables as well as factors associated with the random variables. A factor graph can be defined by a tuple as $F = \langle G, \chi, E \rangle$ where G is a set of vertices denoting random variables, χ is a set of vertices denoting factors, and E is a set of undirected edges connecting the random variables with the factors. When visualizing a factor graph it is common to represent the variables as circles and the factors as squares. It should be noted that factors can be thought

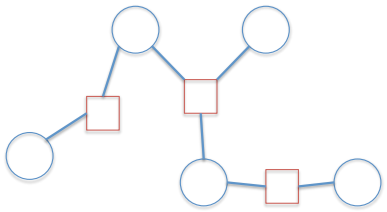

Fig 2.2: Graphical representation of a Factor Graph defined over a joint probability distribution. The circles represent random variables and the squares represent factors associated with the random variables.

of as dependencies between the random variables, though factors can also be associated with a single random variables. Figure 2.2 shows an example of a Factor Graph.

All edges in a Factor Graph are between variable nodes and factor nodes. Factor Graphs make explicit the structure of the factors of the network and have no ambiguities between higher and lower order dependencies between random variables, a property that as we will see Markov Random Fields do not possess. One significant issue with doing so however is that Factor Graphs may require more nodes and complexity than a Markov Random Field would.

Factor Graphs are parameterized by the product of the factors in the graph as

$$P(\mathbf{X}) = \frac{1}{Z} \prod_i \psi_i(\chi_i) \tag{2.4}$$

where $\chi_1, .., \chi_m$ are the set of factors in the model and Z is a normalization constant that ensures the probabilities sum to 1. In most cases a

normalization constant of $Z = \sum_X \prod_i \psi_i(\chi_i)$ is assumed.

2.3 Markov Random Fields

Markov Random Fields (MRFs), like Factor Graphs, are a type of undirected graphical model. An MRF can be defined as a pair $\mathcal{M} = \langle G, \theta \rangle$. Here, $G = (V, E)$ represents an undirected graph where V is a set of vertices representing random variables in a distribution and E is a set of undirected edges connecting those vertices according to the dependencies among the random variables. Just like with Bayesian Networks we can define a "markov blanket" for each random variable, however for MRFs it is much more explicit. The markov blanket $X_{\mathcal{N}(i)}$ for random variable X_i, which is defined as the set of random variables such that $P(X_i \mid \mathbf{X}_{\setminus X_i}) = P(X_i \mid X_{\mathcal{N}(i)})$, is precisely the set of variables with neighboring vertices to the vertex corresponding to X_i (here $\mathcal{N}$ refers to "neighborhood"). Figure 2.3 gives a visual representation of an MRF.

$\Phi = \{\psi_1, .., \psi_m\}$ is a set of potentials associated with an MRF. There are multiple ways of parameterizing an MRF using potentials but the following important theorem tells us that any MRF can be parameterized by a set of potentials corresponding to the maximal cliques in the graph structure.

Theorem 1. **Hammersley-Clifford Theorem:** A distribution $P(\mathbf{X})$ obeys the set of conditional independencies asserted by a Markov Random Field if and only if there exist functions ψ_i such that the distribution can be factorized by $P(\mathbf{X}) = \frac{1}{Z} \prod_c \psi(\mathbf{X}_c)$, where the product is over the set of maximal "cliques"

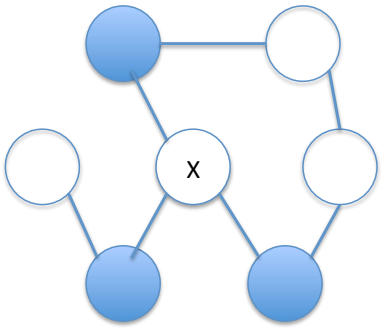

Fig 2.3: Graphical representation of a Markov Random Field (MRF) defined over a joint probability distritbution. The "markov blanket" or "neighborhood" of the variable represented by the X consists of the shaded vertices.

c in the neighborhood graph, and Z is a normalization constant.

By this theorem we get that we can factorize any MRF by a set of potentials with cardinality the same as the number of maximal cliques in the graph structure.

2.3.1 Log Linear Models

In addition to parameterizing an MRF by a set of potentials corresponding to the set of maximum cliques in the graph representation, MRFs can be factorized using a logarithmic representation. Such representations are referred to as *log linear* models.

Definition 12. *A distribution P is a log linear model over a Markov Random Field $\mathcal{M}$ if it can be factorized by a set of features $\mathcal{F} = f_1(D_1), ..., f_k(D_k)$, where each D_i is a complete subgraph over $\mathcal{M}$, each $f_i(D_i)$ is a function $f_i(D_i) : \mathbb{R}^{|D_i|} \rightarrow \mathbb{R}$, and a set of weights $w_1, ..., w_k$ where $w_i \in \mathbb{R}$ such that $P(\mathbf{X}) =$*

$\frac{1}{Z} \exp\left[-\sum_{i=1}^{k} w_i f_i(D_i)\right]$.

In many cases log linear models provide a much more compact and intuitive factorization. For this reason, such a parameterization is often chosen over a standard clique potential factorization.

2.3.2 Ising Model

The *Ising Model* refers to a statistical model that was originally introduced by Wilhelm Lenz in 1920 to represent the energy of a physical system involving a system of interacting atoms. In this model each atom is associated with a binary random variable $X_i \in \{+1, -1\}$. Furthermore, interactions between variables are pairwise and can be parameterized by $\theta_{st}(X_s, X_t) = \theta^*_{st} X_s X_t$ for some parameter $\theta^*_{st} \in \mathbb{R}$ so that the distribution takes the form

$$P_{\theta^*}(\mathbf{X}) = \frac{1}{Z(\theta^*)} \exp\left\{\sum_{(s,t)\in E} \theta^*_{st} X_s X_t\right\} \tag{2.5}$$

From this, we get that the Ising Model can be viewed as a binary, pairwise, Markov Random Field. Many real world distributions can be modeled using binary, pairwise random variables and so the Ising Model has many practical applications.

Chapter 3

Structure Learning

Once we have the structure and parameterization of an MRF we can perform various inferences on our model such as maximum a posteriori (MAP) inference, marginalization, or maximum a posteriori marginal (MPM) inference. However, in many cases we are not given the structure or parameters in advance but instead have access to a set of independent and identically distributed (iid) samples taken from the underlying distribution. In such cases, the problem we are faced with is learning the structure and parameters of the graph given a set of samples. In this paper we will focus primarily on structure learning and will assume that the parameter values can be estimated through other means.

In general, learning the structure of a graphical model from samples is an NP Hard problem and so we will either need to impose specific constraints on the model or assume a level of probabilistic error on any structure learning algorithms [4]. Structure learning can generally be separated into 3 categories of algorithms - constraint based, score based, and regression based. Constraint based algorithms aim at employing conditional independence tests in order to identify a set of conditional independence properties and then attempt

to identify the network structure that best satisfies these constraints [2, 14, 22]. Score based approaches first impose a criterion on the complexity of the graph structure (such as imposed sparsity) and then attempt to search through the space of possible graph structures to determine the model with the highest "score" [7, 8]. Regression based methods attempt to optimize model parameters according to a loss function that often incorporates regularization to impose sparsity on the model in order to reduce overfitting [6, 20, 11].

We will first present existing results on a regression based approach to structure learning that uses ℓ_1-regularized logistic regression to estimate the neighborhood of each node of a pairwise Ising model independently. Next, we will present a new method of structure learning that applies adaptive forward-backward greedy steps to again learn each each node's neighborhood independently. For discrete Markov Random Fields there is no loss of generality in assuming only pairwise interactions since higher-order interactions can be converted to only pairwise interactions by introducing auxilary variables [20]. As we shall see, what separates the greedy approach from the logistic regression method is that rather than solving a multi-variate optimization problem to estimate a node's neighborhood in a single step, efficient greedy steps are performed until the neighborhood converges to a near optimal likelihood. In doing so we are able to estimate the structure of the model with high probability using fewer samples and in less time.

3.1 ℓ_1-Regularized Logistic Regression

ℓ_1 regularized logistic regression has a rich history in fitting linear statistical models to a binary labeled training set of real valued features. In its most general form, given a set of training samples $x_1, .., x_n$ such that $x_i \in \mathbb{R}^p$ and the corresponding labels $y_1, ..., y_n$ such that $y_i \in \{0, 1\}$, the goal of logistic regression is to fit a logistic function $z = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$ so as to minimize some loss (most often the negative log likelihood) that includes a regularization parameter to keep the function from overfitting the training data. More formally, to fit a logistic regression model to a distribution using regularization one can solve the following optimization problem

$$\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} - \log P(y^{(i)} \mid \mathbf{x}^{(i)}; \beta) + \lambda \left\|\beta\right\|_1 \tag{3.1}$$

where $\sum_{i=1}^{n} - \log P(y^{(i)} \mid \mathbf{x}^{(i)}; \beta)$ represents the negative log likelihood and $\lambda \left\|\beta\right\|_1$ represents the regularization portion of the optimization with λ denoting a regularization tuning parameter. As λ is increased it benefits the optimization to set more parameters β_i of the logistic function to 0 and so imposes sparsity on the model. This turns out to be a convex optimization problem and can be solved using modern convex optimization methods.

In estimating the graph structure of of a Markov Random Field, different variations of ℓ_1-regularized logistic regression have been used [6, 15, 20, 11]. In particular, it has been shown that, under an imposed "irrepresentability"

constraint, the graph structure of a pairwise Ising model can be obtained using a sample size of $n = \omega(d^2 \log p)$ with exponentially decaying error by estimating the neighborhood of each node independently by imposing ℓ_1-regularized logistic regression on the node's neighborhood parameters [20]. In the analysis of this method, each node's neighborhood is estimated using the optimization

$$min_{\theta_{\backslash r} \in \mathbb{R}^{p-1}} - \frac{1}{n} \sum_{i=1}^{n} logP_\theta(X_r^{(i)} \mid X_{\backslash r}^{(i)}) + \lambda_{(n,p,d)} \left\| \theta_{\backslash r} \right\|_1 \tag{3.2}$$

where θ is the set of parameters associated with the model and the regularization parameter $\lambda_{(n,p,d)}$ is dependent on the number of samples n, number of nodes p, and the maximum degree of the graph d. The analysis for this method holds true as long as the model satisfies a "dependency condition" that states that the subset of the Fisher information matrix corresponding the the relevant covariates has bounded eigenvalues as well as an "incoherence condition" that states that the large number of irrelevant covariates cannot exert an overly strong effect on the subset of relevant covariates.

3.2 Greedy Neighborhood Selection

Forward-backward greedy algorithms have been applied in a line recent work to statistical estimation of sparse models. These algorithms begin with an initially empty set of parameters and perform both forward steps to greedily add parameters as well as adaptive backward steps to greedily remove parameters whose removal does not increase the loss being optimized by a given

threshold. In particular, T. Zhang [24] analyzed a forward backward greedy algorithm for sparse linear regression and showed that it is sparsistent (consistent for model selection recovery) under the restricted eigenvalue condition. This condition states that no small number (order of the desired sparsity) of features are highly correlated. The novelty of this result is that it is able to achieve sprasistency under a much weaker condition then that of the "irrepresentability" condition required by the traditional Lasso regression approach [23]. In this section we show that using a similar forward backward approach used by Zhang [24] that we can achieve successful parameter estimation in discrete, pairwise Markov Random Fields under weak conditions.

3.2.1 Problem Description

Before we introduce the greedy algorithm that we'll use we must first outline a formal description of the structure learning problem we wish to tackle. Let $\mathbf{X} = (X_1, ..., X_p)$ be a vector consisting of p random variables over some distribution $P(\mathbf{X})$ such that each $X_i \in \{0, 1\}$. A pairwise Markov Random Field over $X = (X_1, ..., X_p)$ is then specified by the pair $\mathcal{M} = \langle G, \Phi \rangle$ where $G = (V, E)$ is a graph over p nodes corresponding to the p variables and Φ is a set of nodewise functions $\theta_r : X \to \mathbb{R}$ for all $r \in V$ along with pariwise functions $\theta_{rt} : X \times X \to \mathbb{R}$ for all $(r, t) \in E$ such that the model can be parameterized by

$$P_{\Phi^*}((X) \propto \exp\left\{\sum_{r\in V}\theta_r(X_r) + \sum_{(r,t)\in E}\theta_{rt}(X_r, X_t)\right\} \tag{3.3}$$

If we focus on the pairwise Ising model which is equivalent to the previous model but imposes an extra constraint that each variable in the distribution is binary according to $X_r \in \{-1, +1\}$ for all $r \in V$ then we can rewrite 3.3 as

$$P_{\Phi^*}(\mathbf{X}) \propto \exp\left\{\sum_{r\in V}\theta_r X_r + \sum_{(r,t)\in E}\theta_{rt} X_r X_t\right\} \tag{3.4}$$

where $\theta_r \in \mathbb{R}$ for all $r \in V$ and $\theta_{rt} \in \mathbb{R}$ for all $(r, t) \in E$.

Let $D = \{X^{(1)}, ..., X^{(n)}\}$ be a set of n samples $X^{(i)} \in \{-1, +1\}^p$ drawn i.i.d. from the distribution P_{Φ^*} defined by parameters Φ^* and graph $G^* = (V, E^*)$. Here, we are using the $*$ superscript to denote the "true" set of parameters and edges. In contrast we will use $\widehat{E}$ to represent "estimated" parameters and edges. The goal of the *graphical model structure learning problem* is to infer the true edge set E^* according the graph $G^* = (V, E^*)$ defined by the Markov Random Field over the distribution $P(\mathbf{X})$ from the sample set $D = \{X^{(1)}, ..., X^{(n)}\}$. In other words, the goal is to construct an estimator $\widehat{E}$ for which $\mathbb{P}\left[\hat{E} = E^*\right] \to 1$ as $n \to \infty$.

3.2.2 Greedy Algorithm for Pairwise Markov Random Fields

We are now ready to introduce the greedy algorithm for structure learning of discrete pairwise Markov Random Fields. An outline of the algorithm is presented in Algorithm 1. The algorithm takes as input a set of samples $Z = \{Z^{(1)}, \ldots, Z^{(n)}\}$ where $Z^{(i)} \in \mathbb{R}^p$, a stopping threshold $\epsilon_s \in \mathbb{R}$, and a backward step factor $v \in \{0, 1\}$. The algorithm works by estimating the neighborhood $\mathcal{N}_r$ of each node independently of the other nodes and finally returns the union of the estimated neighborhoods as the learned set of edges of the graph $\widehat{E} \leftarrow \cup_r \mathcal{N}_r$. In estimating the neighborhood of a fixed node r the algorithm begins with an empty neighborhood $\mathcal{N}_r \leftarrow \emptyset$ and proceeds to greedily choose the "best" neighbor for r and add it to its set of neighbors $\mathcal{N}_r \leftarrow \mathcal{N}_r \cup \{t\}$ as long as the stopping threshold ϵ_s is met. The "best" neighbor is chosen according to the negative log likelihood of the model. In the case of a pairwise Ising model we can define the negative log likelihood in terms of the conditional distribution of X_r given the other variables $X_{\setminus r} = \{X_t \mid t \in V_{\setminus \{r\}}\}$. Referring to [20] we can define the conditional distribution by

$$\mathbb{P}^*_\theta(x_r \mid x_{\setminus r}) = \frac{\exp\left(2x_r \sum_{t \in V_{\setminus r}} \theta^*_{rt} x_t\right)}{\exp\left(2x_r \sum_{t \in V_{\setminus r}} \theta^*_{rt} x_t\right) + 1} \tag{3.5}$$

With this representation of the conditional distribution of our fixed variable X_r we can then define the negative log likelihood as

$$\mathcal{L}_r(\theta; Z_1^n) = -\frac{1}{n}\sum_{i=1}^{n} \log \mathbb{P}_\theta(x_r^{(i)} \mid x_{\setminus r}^{(i)}) \tag{3.6}$$

Then, in our forward step of choosing the "best" neighbor of X_r we optimize over 3.6 by

$$\arg\min_{t \in V_{\setminus r}} \min_{\theta_{rt}} \mathcal{L}_r(\theta \cup \{\theta_{rt}\}; Z_1^n) \tag{3.7}$$

After choosing the "best" neighbor for r according to 3.7, the algorithm then compares the loss before the neighbor t was added to $\mathcal{N}_r$, $\mathcal{L}_r(\theta; Z_1^n)$, to the loss after adding t, $\mathcal{L}_r(\theta \cup \{\theta_{rt}\}; Z_1^n)$, and adds the edge as long as adding it reduces the loss by a factor of ϵ_s, otherwise the stopping criterion is met and the algorithm terminates. After each successful forward step the algorithm first optimizes over the current parameters in the neighborhood $\mathcal{N}_r$ as adding an edge may have affected the optimal neighborhood parameter values. Next, the algorithm performs a backward step in which it checks the *influence* of all variables in the current neighborhood $\mathcal{N}_r$ in the presence of the newly added variable from the forward step. If one of more of the previously added variables do not contribute at least $v\epsilon_s$ to the loss function, then the algorithm removes them from the neighborhood $\mathcal{N}_r$. This procedure ensures that at each round, the loss function is improved by at least $(1 - v)\epsilon_s$ and hence it terminates within a finite number of steps.

In order to guarantee sparsistency, the pairwise greedy algorithm requires the conditions of *restricted strong convexity* and *restricted strong smoothness* on the negative log likelihood in terms of the graph we're trying to estimate. In terms of general statistical models, Neghaban et al. [17] define restricted convexity in terms of a general loss function as

Definition 13. Restricted Strong Convexity: *Given a set* $\mathbb{S}$ *and a set of samples* $\mathbf{Z} = \{Z_1, \ldots, Z_n\}$*, a loss function* $\mathcal{L}(.)$ *is said to satisfy restricted strong convexity (RSC) with parameter* k_l *if for all* $\Delta \in \mathbb{S}$ *such that* $|\Delta| \leq k$ *we get that* $\mathcal{L}(\theta + \Delta; Z_1^n) - \mathcal{L}(\theta; Z_1^n) - \langle \nabla \mathcal{L}(\theta; Z_1^n), \Delta \rangle \geq \frac{k_l}{2} \|\Delta|_2^2$

Since we are assuming our distribution is sparse we will also introduce syntactic sugar to reference the non-zero parameters of the model's parameterization. Let $S \subseteq \{1, ..., p\}$ denote the support set for the parameterization of a fixed node $r \in V$. That is, S is the set of indices of the neighborhood $\widehat{\mathrm{N}}_r$ of r where $\theta_{rt} \neq 0$. Given our general definition of RSC, we can define sparsity restricted strong convexity as

Definition 14. Sparsity Restricted Strong Convexity: *Given a set of samples* $\mathbf{Z} = \{Z_1, \ldots, Z_n\}$*, a loss function* $\mathcal{L}(.)$ *is said to satisfy sparsity restricted strong convexity (RSC(k)) with parameter* k_l *if for all sets* $S \subseteq \{1, \ldots, p\}$ *such that* $|S| \leq k$*,* $\mathcal{L}(.)$ *satisfies RSC with parameter* k_l *where* S *is the support set on* θ*.*

The definition for RSC(k) is defined in terms of a loss function satisfying RSC for all support sets $S \subseteq \{1, \ldots, p\}$ over the set of variables in the

distribution such that $|S| \leq k$. In other words, a loss function satisfies RSC(k) if it satisfies RSC for all sets of parameters θ where $\|\theta\|_0 \leq k$.

In contrast to RSC we can define restricted strong smoothness (RSS) in terms of a general loss function as

Definition 15. Restricted Strong Smoothness: *Given a set $\mathbb{S}$ and a set of samples $\mathbf{Z} = \{Z_1, \ldots, Z_n\}$, a loss function $\mathcal{L}(.)$ is said to satisfy restricted strong smoothness (RSS) with parameter k_u if for all $\Delta \in \mathcal{S}$ such that $|\Delta| \leq k$ we get that $\mathcal{L}(\theta + \Delta; Z_1^n) - \mathcal{L}(\theta; Z_1^n) - \langle \nabla \mathcal{L}(\theta; Z_1^n), \Delta \rangle \leq \frac{k_u}{2} \|\Delta|_2^2$*

Additionally, we can define sparsity restricted strong smoothness in terms of a general loss as

Definition 16. Sparsity Restricted Strong Smoothness: *a set of samples $\mathbf{Z} = \{Z_1, \ldots, Z_n\}$, a loss function $\mathcal{L}(.)$ is said to satisfy restricted strong smoothness (RSS(k)) with parameter k_u if for all sets $S \subseteq \{1, \ldots, p\}$ such that $|S| \leq k$, $\mathcal{L}(.)$ satisfies RSS with parameter k_l where S is the support set on θ.*

Another property that our analysis requires is an upper bound λ_n on the ℓ_∞ norm of the gradient of the negative log likelihood at the true parameter θ^*. That is, for each $r \in V$ we require a bound $\lambda_n \geq \|\nabla \mathcal{L}_r(\theta^*)\|_\infty$. This captures the "noise level" of the samples with respect to the loss.

Before we state our main theorem, we first need to state some auxiliary lemmas that capture the sparsistecy of the forward and backward steps in Algorithm 1.

3.2.2.1 Lemmas for Main Theorem

We list the simple lemmas that characterize the parameter $\widehat{\theta}$ obtained when the inner while loop of the algorithm returns, and on which the proof of our main theorem depends. The proofs of these lemmas can be found in the Appendix.

Lemma 1 (Stopping Forward Step)**.** When the inner while loop (forward step) of algorithm 1 returns with parameter $\widehat{\theta}$ supported on $\widehat{S}$, we have

$$\left|\mathcal{L}\left(\widehat{\theta}\right) - \mathcal{L}\left(\theta^*\right)\right| < \sqrt{2\left|S^* - \widehat{S}\right|\kappa_u\,\epsilon_{\mathcal{S}}}\ \left\|\widehat{\theta} - \theta^*\right\|_2.$$

Lemma 2 (Stopping Error Bound)**.** When the inner while loop (forward step) of algorithm 1 returns with parameter $\widehat{\theta}$ supported on $\widehat{S}$, we have

$$\left\|\widehat{\theta} - \theta^*\right\|_2 \leq \frac{2}{\kappa_l}\left(\lambda_n\sqrt{\left|S^* \cup \widehat{S}\right|} + \sqrt{2\left|S^* - \widehat{S}\right|\kappa_u\epsilon_{\mathcal{S}}}\right).$$

Lemma 3 (Stopping Backward Step)**.** When the inner while loop (forward step) of algorithm 1 returns with parameter $\widehat{\theta}$ supported on $\widehat{S}$, we have

$$\left\|\widehat{\Delta}_{\widehat{S}-S^*}\right\|_2^2 \geq \frac{\epsilon_{\mathcal{S}}}{\kappa_u}\left|\widehat{S} - S^*\right|.$$

Lemma 4 (Stopping Size)**.** If $\epsilon_{\mathcal{S}} > \frac{\lambda_n^2}{\kappa_u}\left(\sqrt{\frac{2}{\eta-1}} - \sqrt{\frac{2}{\eta}}\right)^{-2}$ and $RSC\left(\eta s^*\right)$ holds for some $\eta \geq 2 + 4\rho^2\left(\sqrt{\frac{\rho^2-\rho}{s^*}} + \sqrt{2}\right)^2$, then the inner while loop (forward of algorithm 1 returns with $k \leq (\eta - 1)s^*$.

Notice that if $\epsilon_{\mathcal{S}} \geq (8\rho\eta/\kappa_l)\ (\eta^2/(4\rho^2))\ \lambda_n^2$, then, the assumption of this lemma is satisfied. Hence for large value of $s^* \geq 8\rho^2 > \eta^2/(4\rho^2)$, it suffices to have $\epsilon_{\mathcal{S}} \geq (8\rho\eta/\kappa_l)\ s^*\lambda_n^2$.

3.2.2.2 Main Theorem

We are now ready to state the our main theorem on the sparsitency of the pairwise greedy algorithm. Suppose we are given a set of n i.i.d. samples $Z = \{Z_1, ..., Z_n\}$ where $Z_i \in \{-1, +1\}^p$, drawn from the distribution according to a pairwise Ising model as in 3.4, with parameters θ^* and graph $G = (V, E^*)$. Let the maximum degree of the graph G be denoted by d.

Theorem 2 (Pairwise Sparsistency). Suppose we run Algorithm 1 with stopping threshold $\epsilon_{\mathbb{S}} \geq c_1 \frac{d \log p}{n}$, where, d is the maximum node degree in the graphical model, and the true parameters θ^* satisfy $\frac{c_3}{\sqrt{d}} > \min_{j \in S^*} |\theta^*_j| > c_2 \sqrt{\epsilon_{\mathbb{S}}}$, and further that number of samples scales as

$$n > c_4 \, d^2 \, \log p,$$

for some constants c_1, c_2, c_3, c_4. Then, with probability at least $1 - c' \exp(-c'' n)$, the output $\widehat{\theta}$ supported on $\widehat{S}$ satisfies:

(a) **No False Exclusions:** $E^* - \widehat{E} = \emptyset$.

(b) **No False Inclusions:** $\widehat{E} - E^* = \emptyset$.

Proof. We first show that our assumptions hold under the negative log likelihood loss function that Algorithm 1 uses. Here, we use arguments taken from [10].

RSC, RSS. First, we note that the conditional log-likelihood loss function in (3.6) corresponds to a logistic likelihood. Moreover, the covariates are all

binary, and bounded, and hence also sub-Gaussian. [17, 1] analyze the RSC and RSS properties of generalized linear models, of which logistic models are an instance, and show that the following result holds if the covariates are sub-Gaussian. Let $\partial\mathcal{L}(\Delta;\theta^*) = \mathcal{L}(\theta^*+\Delta) - \mathcal{L}(\theta^*) - \langle\nabla\mathcal{L}(\theta^*),\Delta\rangle$ be the second order Taylor series remainder. Then, Proposition 2 in [17] states that that there exist constants κ_1^l and κ_2^l, independent of n, p such that with probability at least $1 - c_1\exp(-c_2 n)$, for some constants $c_1, c_2 > 0$,

$$\partial\mathcal{L}(\Delta;\theta^*) \;\geq\; \kappa_1^l\|\Delta\|_2\left\{\|\Delta\|_2 - \kappa_2^l\sqrt{\frac{\log(p)}{n}}\|\Delta\|_1\right\} \qquad \text{for all } \Delta : \|\Delta\|_2 \leq 1.$$

Thus, if $\|\Delta\|_0 \leq k := \eta d$, then $\|\Delta\|_1 \leq \sqrt{k}\|\Delta\|_2$, so that

$$\partial\mathcal{L}(\Delta;\theta^*) \;\geq\; \|\Delta\|_2^2\left(\kappa_1^l - \kappa_2^l\sqrt{\frac{k\log p}{n}}\right) \;\geq\; \frac{\kappa_1^l}{2}\|\Delta\|_2^2,$$

if $n > 4(\kappa_2^l/\kappa_1^l)^2\,\eta d\,\log(p)$. In other words, with probability at least $1 - c_1\exp(-c_2 n)$, the loss function $\mathcal{L}$ satisfies $RSC(k)$ with parameter κ_1^l provided $n > 4(\kappa_2^l/\kappa_1^l)^2\,\eta d\,\log(p)$. Similarly, it follows from [17, 1] that there exist constants κ_1^u and κ_2^u such that with probability at least $1 - c_1'\exp(-c_2' n)$,

$$\partial\mathcal{L}(\Delta;\theta^*) \;\leq\; \kappa_1^u\|\Delta\|_2\{\|\Delta\|_2 - \kappa_2^u\|\Delta\|_1\} \qquad \text{for all } \Delta : \|\Delta\|_2 \leq 1,$$

so that by a similar argument, with probability at least $1 - c_1'\exp(-c_2' n)$, the loss function $\mathcal{L}$ satisfies $RSS(k)$ with parameter κ_1^u provided $n > 4(\kappa_2^u/\kappa_1^u)^2\,\eta d\,\log(p)$.

Noise Level. Next, we obtain a bound on the noiselevel $\lambda_n \geq \|\nabla\mathcal{L}(\theta^*)\|_\infty$ following similar arguments to [20]. Let W denote the gradient $\nabla\mathcal{L}(\theta^*)$ of the loss function (3.6). Any entry of W has the form $W_t = \frac{1}{n}\sum_{i=1}^n Z_{rt}^{(i)}$, where

$Z_{rt}^{(i)} = x_t^{(i)}(x_r^{(i)} - \mathbb{P}(x_r = 1|x_{\setminus s}^{(i)}))$ are zero-mean, i.i.d. and bounded $|Z_{rt}^{(i)}| \leq 1$. Thus, an application of Hoeffding's inequality yields that $\mathbb{P}[|W_t| > \delta] \leq 2\exp(-2n\delta^2)$. Applying a union bound over indices in W, we get $\mathbb{P}[\|W\|_\infty > \delta] \leq 2\exp(-2n\delta^2 + \log(p))$. Thus, if $\lambda_n = (\log(p)/n)^{1/2}$, then $\|W\|_\infty \leq \lambda_n$ with probability at least $1 - \exp(-n\lambda_n^2 + \log(p))$.

We have now shown that the conditions of RSC, RSS, and "bounded noise level" all hold with respect to the negative log likelihood loss. To prove the main result of our theorem we will first show that each node neighborhood returned after the inner while loop (forward step) returns satisfies $S^* - \widehat{S} = \emptyset$ and $\widehat{S} - S^* = \emptyset$ and then we will use a simple union bound to show that the returned edge set $\widehat{E}$ also satisfies these conditions.

No False Exclusions ($S^* - \widehat{S} = \emptyset$). We use a chaining argument similar to that used in [10] and [24]. For any $\tau \in \mathbb{R}$, we have

$$\begin{aligned}\tau|\{j \in S^* - \widehat{S} : |\theta_j^*|^2 > \tau\}| &\leq \|\theta^*_{S^* - \widehat{S}}\|_2^2 \leq \|\theta^* - \widehat{\theta}\|_2^2 \\ &\leq \frac{8\eta s^* \lambda_n^2}{\kappa_l^2} + \frac{16\kappa_u \epsilon_{\mathcal{S}}}{\kappa_l^2}|S^* - \widehat{S}|,\end{aligned}$$

where the last inequality follows from part (a) and the inequality $(a+b)^2 \leq 2a^2 + 2b^2$. Now, setting $\tau = \frac{32\kappa_u \epsilon_{\mathcal{S}}}{\kappa_l^2}$, and dividing both sides by $\tau/2$ we get

$$2|\{j \in S^* - \widehat{S} : |\theta_j^*|^2 > \tau\}| \leq \frac{\eta s^* \lambda_n^2}{2\kappa_u \epsilon_{\mathcal{S}}} + |S^* - \widehat{S}|.$$

Substituting $|\{j \in S^* - \widehat{S} : |\theta_j^*|^2 > \tau\}| = |S^* - \widehat{S}| - |\{j \in S^* - \widehat{S} : |\theta_j^*|^2 \leq \tau\}|$, we get

$$|S^* - \widehat{S}| \leq |\{j \in S^* - \widehat{S} : |\theta_j^*|^2 \leq \tau\}| + \frac{\eta s^* \lambda_n^2}{2\kappa_u \epsilon_{\mathcal{S}}} \leq |\{j \in S^* - \widehat{S} : |\theta_j^*|^2 \leq \tau\}| + 1/2,$$

due to the setting of the stopping threshold $\epsilon_{\mathcal{S}}$. This in turn entails that

$$|S^* - \widehat{S}| \leq |\{j \in S^* - \widehat{S} : |\theta_j^*|^2 \leq \tau\}| = 0,$$

by our assumption on the size of the minimum entry of θ^*.

No False Inclusions ($\widehat{S} - S^* = \emptyset$). Using an argument similar to that used in [10], by Lemma 3, which provides a simple consequence of the backward step failing when the inner while loop (forward step) returns, for $\widehat{\Delta} = \widehat{\theta} - \theta^*$, we have $\epsilon_{\mathrm{S}}/\kappa_u|\widehat{S} - S^*| \leq \|\widehat{\Delta}_{\widehat{S}-S^*}\|_2^2 \leq \|\widehat{\Delta}\|_2^2$, so that using Lemma 2 and that $|S^* - \widehat{S}| = 0$, we obtain that $|\widehat{S} - S^*| \leq \frac{4\eta s^* \lambda_n^2 \kappa_u}{\epsilon_{\mathrm{S}} \kappa_l^2} \leq 1/2$, due to the setting of the stopping threshold ϵ_{S}.

Then, using a union bound on the node neighborhoods $\mathcal{N}_r$ obtained from the inner while loop (forward step) we are able to prove our intended result that $\widehat{E} - E^* = \emptyset$ and $E^* - \widehat{E} = \emptyset$ with probability at least $1 - c' \exp(-c'' n)$.

□

The sufficient conditions imposed on the edge parameters by the pairwise greedy algorithm is a restricted strong convexity condition [17], which is weaker than the irrepresentability condition required by [20]. Furthermore, the number of samples required for sparsistent graph recovery scales as $O(d \log p)$, where d is the maximum node degree, in contrast to $O(d^2 \log p)$ for ℓ_1-regularized logistic regression. We further corroborate these results in our empirical simulations, where we find that the greedy algorithm requires fewer observations than [20] for sparsistent graph recovery.

Algorithm 1 Greedy forward-backward algorithm for pairwise discrete graphical model structure learning

Input: Samples $Z = \{Z^{(1)}, \ldots, Z^{(n)}\}$, Stopping Threshold ϵ_S, Backward Step Factor $\nu \in (0,1)$

Output: Estimated Edges $\widehat{E}$

for $r \in V$ **do**

let $\widehat{\theta}^{(0)} \leftarrow \emptyset$, $\widehat{N}_r^{(0)} \leftarrow \emptyset$, and $k = 0$

while true **do** {*Forward Step*}

$(j_*, \alpha_*) \longleftarrow \arg \min_{j \in \left(\widehat{\mathbb{N}}_r^{(k-1)}\right)^c; \alpha} \mathcal{L}(\widehat{\theta}^{(k-1)} + \alpha e_j; Z_1^n)$

$\widehat{\mathbb{N}}_r^{(k)} \longleftarrow \widehat{\mathbb{N}}_r^{(k-1)} \cup \{j_*\}$

$\delta_f^{(k)} \longleftarrow \mathcal{L}(\widehat{\theta}^{(k-1)}; Z_1^n) - \mathcal{L}(\widehat{\theta}^{(k-1)} + \alpha_* e_{j_*}; Z_1^n)$

if $\delta_f^{(k)} \leq \epsilon_{\mathcal{S}}$ **then**

break

end if

$\widehat{\theta}^{(k)} \longleftarrow \arg\min_{\theta} \mathcal{L}\left(\theta_{\widehat{\mathbb{N}}_r^{(k)}}; Z_1^n\right)$

$k \longleftarrow k + 1$

while true **do** {*Backward Step*}

$j^* \longleftarrow \arg \min_{j \in \widehat{\mathbb{N}}_r^{(k-1)}} \mathcal{L}(\widehat{\theta}^{(k-1)} - \widehat{\theta}_j^{(k-1)} e_j; Z_1^n)$

if $\mathcal{L}\left(\widehat{\theta}^{(k-1)} - \widehat{\theta}_{j^*}^{(k-1)} e_{j^*}; Z_1^n\right) - \mathcal{L}\left(\widehat{\theta}^{(k-1)}; Z_1^n\right) > \nu \delta_f^{(k)}$ **then**

break

end if

$\widehat{\mathbb{N}}_r^{(k-1)} \longleftarrow \widehat{\mathbb{N}}_r^{(k)} - \{j^*\}$

$\widehat{\theta}^{(k-1)} \longleftarrow \arg\min_{\theta} \mathcal{L}\left(\theta_{\widehat{\mathbb{N}}_r^{(k-1)}}; Z_1^n\right)$

$k \longleftarrow k - 1$

end while

end while

end for

Output $\widehat{E} = \bigcup_r \left\{(r,t) : t \in \widehat{\mathbb{N}_r}\right\}$

Chapter 4

Experimentation

In this section we present experimental results that illustrate the power of Algorithm 1 as well as support our theoretical guarantee that the algorithm fully achieves graph selection for samples scaling as $n = \Omega(d \log(p))$. We first compare Algorithm 1 to that of the node wise ℓ_1-regularized logistic regression method outlined by Ravikumar et al. in [20] using simulated data on 3 different graph structures. We then revisit the music recommendation question posed in the introduction of this document by fitting a pairwise Ising model to a real world music listening dataset using the pairwise greedy algorithm.

4.1 Simulated Analysis

In comparing the pairwise greedy algorithm to that of ℓ_1-regularized logistic regression we performed experiments using 3 different graph structures: (a) chain (line graph), (b) 4-nearest neighbor grid, and (c) star graph. A visual representation of each graph type is outlined in 4.1. For each graph type we simulated structure learning for graph sizes of $p = 36$, $p = 64$, and $p = 100$ nodes using both the pairwise greedy algorithm as well as node wise ℓ_1-regularized logistic regression. For each experiment, we built a simulated graph

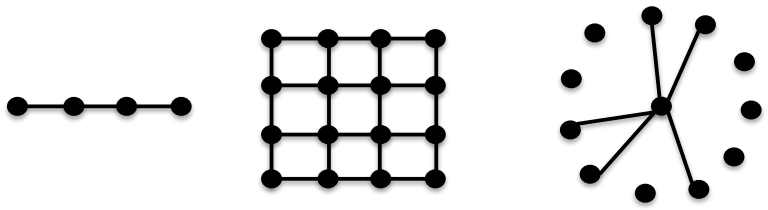

Fig 4.1: Chain, 4-Nearest Neighbor Grid, and Star Graphs

of the corresponding type with random mixed sign edges $\theta_{rt} \in \{-0.5, +0.5\}$. We then generated sets of samples $x^{(1)}, \ldots, x^{(n)}$ from each model using Gibbs sampling and ran each algorithm using the samples as input. We then compared the empirically learned edge set $\widehat{E}$ to that of the true edge set E^*. If the edge sets matched completely ($\widehat{E} = E^*$) then we declared the result a *success* and otherwise declared it a *failure*. Using a batch size of 10 randomly generated models and samples (using Gibbs sampling) we averaged the probability of "success" and scaled the number of samples n until $\mathbb{P}(\widehat{E} = E^*) \to 1$. For all greedy experiments we used a stopping threshold of $\epsilon_s = \frac{c \log(np)}{n}$ where c is a tuning constant, as suggested by Theorem 2, and set the backwards step threshold $v = 0.5$. For all ℓ_1-logistic regression experiments we used a regularization parameter of $\lambda_n = c' \sqrt{\log(p)/n}$, where c' was set via cross-validation according to [20].

Figures 4.2, 4.3, and 4.4 show the results for the chain ($d = 2$), grid ($d = 4$) and star ($d = 0.1p$) graphs using both Algorithm 1 and ℓ_1-logistic regression for three different graph sizes $p \in \{36, 64, 100\}$ with random, mixed sign ($\theta_{rt} \in \{-0.5, +0.5\}$) edge couplings. For each sample size, we generated a batch of 10 different graphical models and averaged the probability of success

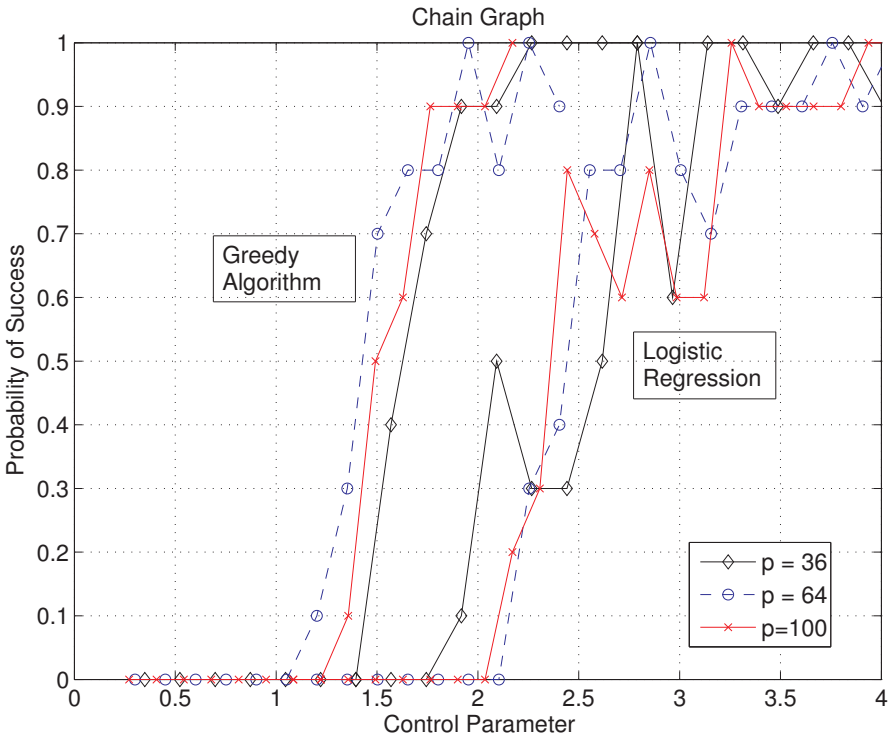

Fig 4.2: Chain Plot

(complete structure learned) over the batch. Each curve then represents the probability of success versus the control parameter $\beta(n, p, d) = n/[20d\log(p)]$ which increases with the sample size n. These results support our theoretical claims and demonstrate the efficiency of the greedy method in comparison to logistic regression [20].

4.2 Real Data Analysis

Let us now return to the question we posed at the beginning of this document. Suppose we were charged with the task of predicting new music

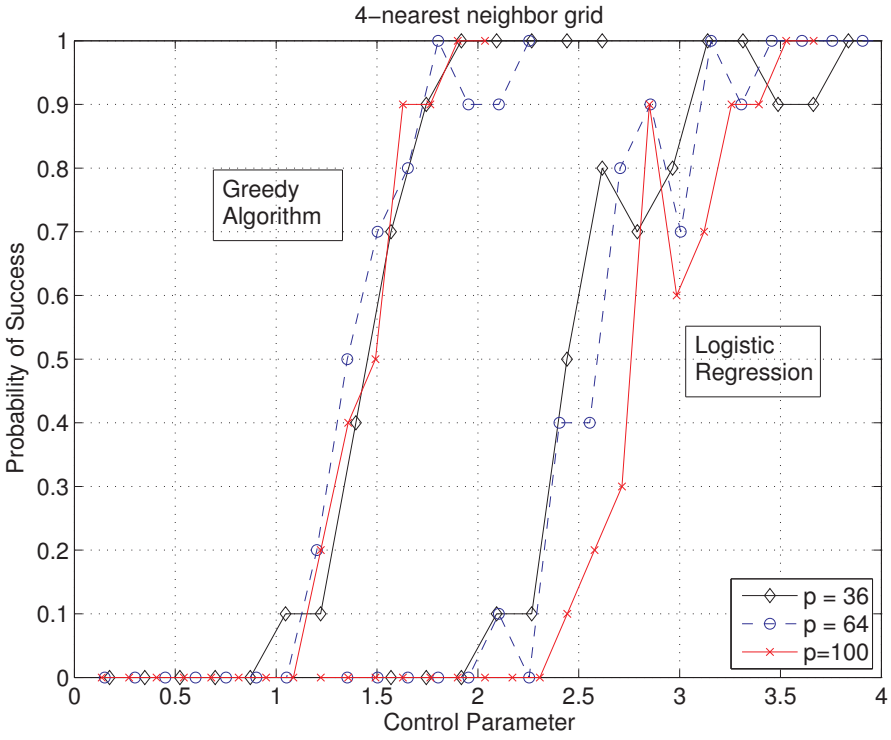

Fig 4.3: Grid Plot

that a listener may enjoy given music that they currently listen to. It is of course natural to think that there are dependencies between music artists and listeners listening habits. In fact, we can view the set of music artists that a listener may choose to listen to as a large pairwise Ising model. We let the set of music artists that we are considering each be represented by a node in our graph. Then, we can assume that the edges of our graph represent dependencies among the artists. A single sample of our distribution would then represent the artists that a specific listener likes $(+1)$ as well as the artists that they do not necessarily like (-1). We can assume that dependences between

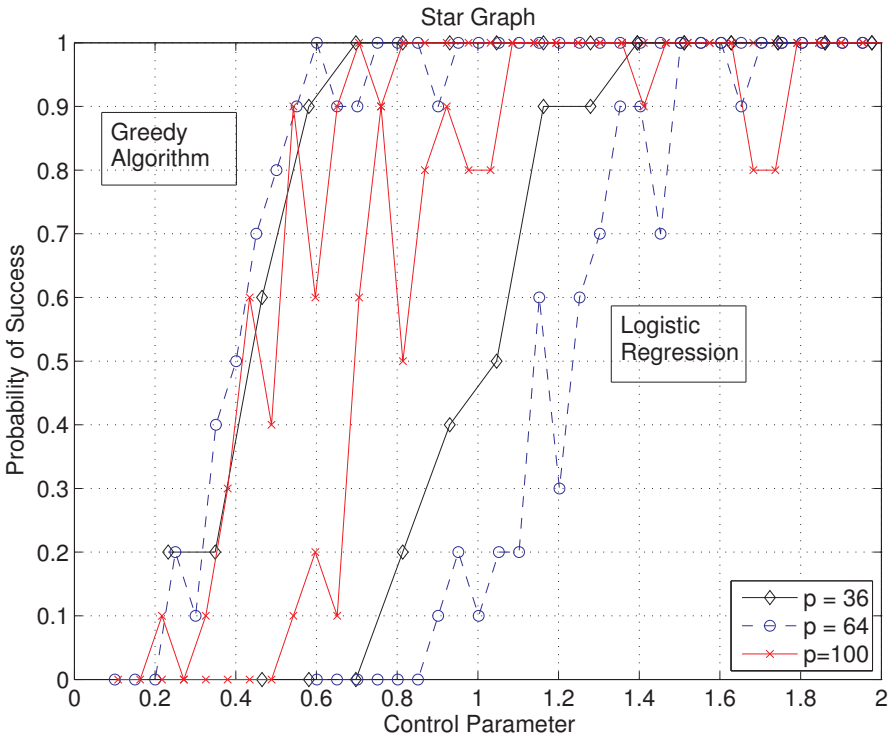

Fig 4.4: Star Plot

artists may be both *positve* ($\theta_{rt} > 0$), meaning that if a listener likes one artist then they are likely to like the other, as well as *negative* ($\theta_{rt} < 0$), meaning that if a listener likes one artist then they are likely to not like the other artist.

In an effort both to test our basic assumptions regarding music preferences fitting a pairwise Ising model as well as to experiment using the pairwise greedy structure learning algorithm we attempted to fit a pairwise Ising model to the Audioscrobbler music listener dataset [9]. Audioscrobbler, which has since merged with music radio website last.fm, was a popular music player plugin that tracked music listening habits for its users for use with analytics

and collaborative filtering. The Audioscrobbler datatset [9] is a freely available dataset containing profiles for over $150,000$ users, taken over a period of several months. For each user in the dataset the data lists which artists the user listened to as well as a counter indicating how many times they listened to each artist over the recorded period. For our experimentation, we chose 20 music artists to investigate, which included 11 modern *indie* rock bands and 9 *classic* rock bands. We chose to only include users who had listened to at least 3 artists from our model so as to decrease the sparsity of the resulting model. We then generated a set of samples $x^{(1)}, \ldots, x^{(n)}$ where each $x^{(i)} \in \{0,1\}^{20}$ corresponding to each of the users in our dataset that had listened to at least 3 of the artists from our model. In generating the samples we used a listening threshold $\lambda = 20$ to determine the values of the samples. If a user i had listened to an artist r more than λ times then we set $x_r^{(i)} = +1$ otherwise we set $x_r^{(i)} = -1$. Altogether, we generated a set of over $14,000$ samples ($14,000$ different users). Then, using these samples we ran the pairwise greedy algorithm to learn the structure of the underlying graph. After learning the structure of the graph we kept all positive edges and discarded all edges for which the algorithm decided were negative (since negative correlations would not be helpful in predicting new music). A visual representation of the learned positive graph structure is shown in Figure 4.5. In addition, an adjacency matrix representation of the learned positive graph structure is shown in Figure 4.2. These results show that intuitive music artist dependencies were accurately detected using the pairwise greedy structure learning

algorithm. For example, Neil Young was found to have positive dependence with Jimi Hendrix, Simon & Garfunkel, Bob Dylan, Velvet Undergound, Janis Joplin, and Van Morrison, each of which share a similar style and demographic to that of Neil Young. In order to obtain a more comprehensive measure of success we also decided to compare our results to the "similar artists" section of music website AllMusic.com [3]. Here, we explicitly assumed that the "similar artists" section of [3] was composed by music experts and thus accurately reflects truthful music dependencies. In comparing our results to those of [3] we obtained

$$\text{Precision}: \ 0.45 \tag{4.1}$$

$$\text{Recall}: \ 0.67 \tag{4.2}$$

$$\text{F}_1: \ 0.54 \tag{4.3}$$

Although it is difficult to accurately annotate truth in music dependencies these are rather positive results that demonstrate the pairwise greedy algorithm's ability to successfully learn the structure of a pairwise Ising model using real data.

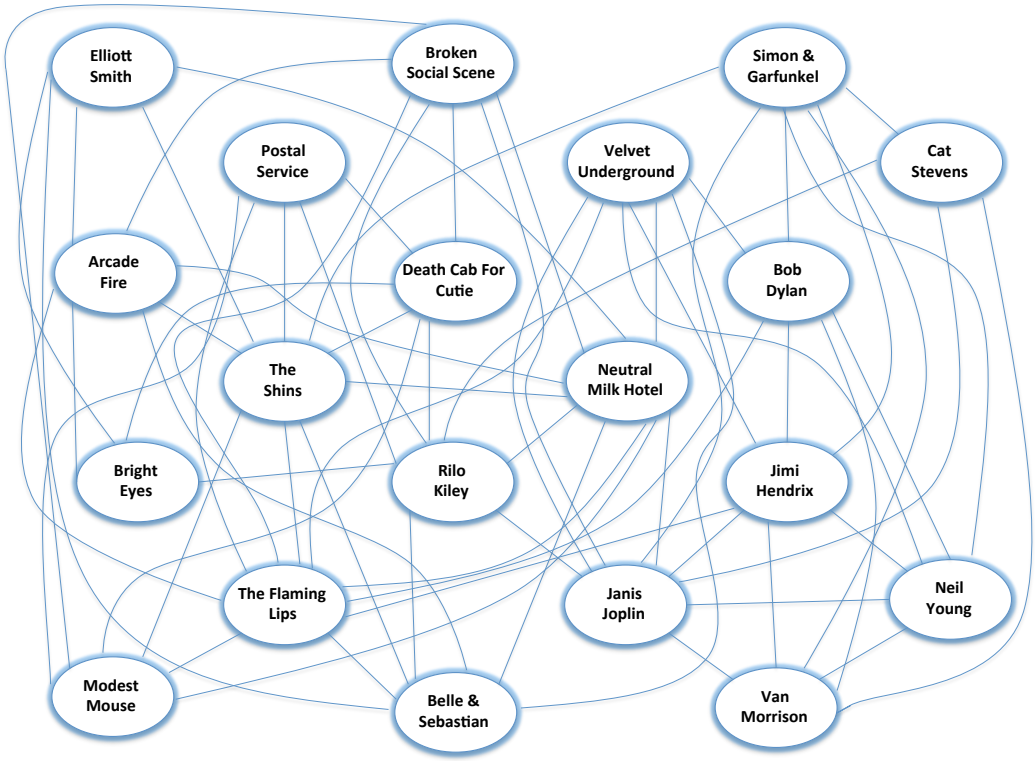

Fig 4.5: Visual graph representation of music preferences MRF structure.

	Neil Young	Broken Social Scene	Cat Stevens	Death Cab For Cutie	Neutral Milk Hotel	Arcade Fire	Belle and Sebastian	Elliott Smith	Postal Service	Rilo Kiley	Jimi Hendrix	The Shins	Flaming Lips	Bright Eyes	Modest Mouse	Simon and Garfunkel	Bob Dylan	Velvet Underground	Janis Joplin	Van Morrison
Neil Young											X					X	X	X	X	X
Broken Social Scene				X	X	X				X	X			X		X			X	
Cat Stevens										X						X			X	X
Death Cab For Cutie		X							X	X		X		X	X					
Neutral Milk Hotel	X					X	X	X		X		X	X		X			X	X	
Arcade Fire		X			X		X	X				X	X	X						
Belle and Sebastian					X	X		X		X		X	X					X		
Elliott Smith					X	X	X					X		X						
Postal Service				X						X		X	X		X					
Rilo Kiley		X	X	X	X		X		X					X					X	
Jimi Hendrix	X												X			X	X	X	X	X
The Shins				X	X	X	X	X	X				X		X	X				
Flaming Lips		X			X	X	X		X		X	X			X		X	X		
Bright Eyes				X		X		X		X										
Modest Mouse		X		X	X				X			X	X							
Simon and Garfunkel	X		X								X	X					X		X	X
Bob Dylan	X										X		X			X		X		X
Velvet Underground	X				X		X				X		X				X		X	
Janis Joplin	X	X	X		X					X	X					X		X		X
Van Morrison	X		X								X					X	X		X	

Fig 4.6: Adjacency matrix representation of music preferences MRF stucture.

Appendices

Appendix A

Proofs of Auxiliary Lemmas

The proofs of Lemmas 1, 2, and 3 listed here are reprinted from [10].

Note that when the inner while loop (forward step) in the greedy algorithm returns, the forward step fails to go through. This entails that

$$\mathcal{L}(\widehat{\theta}) - \inf_{j \in \widehat{S}^c, \alpha \in \mathbb{R}} \mathcal{L}(\widehat{\theta} + \alpha e_j) < \epsilon_{\mathcal{S}}. \tag{A.1}$$

The next lemma shows that this has the consequence of upper bounding the deviation in loss between the estimated parameters $\widehat{\theta}$ and the true parameters θ^*.

Lemma 1 (Stopping Forward Step). When the inner while loop (forward step) returns with parameter $\widehat{\theta}$ supported on $\widehat{S}$, we have

$$\left| \mathcal{L}\left(\widehat{\theta}\right) - \mathcal{L}\left(\theta^*\right) \right| < \sqrt{2\left|S^* - \widehat{S}\right| \kappa_u \, \epsilon_{\mathcal{S}}} \, \left\| \widehat{\theta} - \theta^* \right\|_2. \tag{A.2}$$

Proof. Let $\widehat{\Delta} = \theta^* - \widehat{\theta}$. For any $\eta \in \mathbb{R}$, we have

$$\mathcal{L}\left(\widehat{\theta} + \eta \widehat{\Delta}_j e_j\right) \leq \mathcal{L}\left(\widehat{\theta}\right) + \eta \nabla_j \mathcal{L}\left(\widehat{\theta}\right) \widehat{\Delta}_j + \eta^2 \frac{\kappa_u}{2} \widehat{\Delta}_j^2.$$

Thus, we can establish

$$-|S^* - \widehat{S}|\epsilon_{\mathcal{S}} < \sum_{j \in S^* - \widehat{S}} \left(\mathcal{L}\left(\widehat{\theta} + \eta \widehat{\Delta}_j e_j\right) - \mathcal{L}\left(\widehat{\theta}\right) \right)$$
$$\leq \eta \Big(\mathcal{L}\left(\theta^*\right) - \mathcal{L}\left(\widehat{\theta}\right) \Big) + \eta^2 \frac{\kappa_u}{2} \left\| \widehat{\Delta} \right\|_2^2 .$$

Optimizing the RHS over η, we obtain

$$-|S^* - \widehat{S}|\, \epsilon_{\mathcal{S}} < - \frac{\left(\mathcal{L}(\theta^*) - \mathcal{L}\left(\widehat{\theta}\right) \right)^2}{2\, \kappa_u \, \|\widehat{\Delta}\|_2^2},$$

whence the lemma follows.

□

Lemma 2 (Stopping Error Bound). When the inner while loop (forward step) returns with parameter $\widehat{\theta}$ supported on $\widehat{S}$, we have

$$\|\widehat{\theta} - \theta^*\|_2 \leq \frac{2}{\kappa_l} \left(\lambda_n \sqrt{\left| S^* \cup \widehat{S} \right|} + \sqrt{2 \left| S^* - \widehat{S} \right| \kappa_u \epsilon_{\mathcal{S}}} \right). \tag{A.3}$$

Proof. For $\Delta \in \mathbb{R}$, let

$$G(\Delta) = \mathcal{L}\left(\theta^* + \Delta\right) - \mathcal{L}\left(\theta^*\right) - \sqrt{2 \left| S^* - \widehat{S} \right| \, \kappa_u \, \epsilon_{\mathcal{S}}} \, \|\Delta\|_2 .$$

It can be seen that $G(0) = 0$, and from the previous lemma, $G(\widehat{\Delta}) \leq 0$. Further, $G(\Delta)$ is sub-homogeneous (over a limited range): $G(t\Delta) \leq tG(\Delta)$ for $t \in [0, 1]$. Thus, for a carefully chosen $r > 0$, if we show that $G(\Delta) > 0$ for all $\Delta \in \{\Delta : \|\Delta\|_2 \leq r,\ \|\Delta\|_0 \leq |S|\}$, where $S = |\widehat{S} \cup S^*|$, then it follows that $\|\widehat{\Delta}\|_2 \leq r$. If not, then there would exist some $t \in [0, 1)$ such that $\|t\widehat{\Delta}\| = r$, whence we would arrive at the contradiction

$$0 < G(t\widehat{\Delta}) \leq tG(\widehat{\Delta}) \leq 0.$$

Thus, it remains to show that $G(\Delta) > 0$ for all $\Delta \in \{\Delta : \|\Delta\|_2 \leq r,\ \|\Delta\|_0 \leq |S|\}$. By restricted strong convexity property of $\mathcal{L}$, we have

$$\mathcal{L}(\theta^* + \Delta) - \mathcal{L}(\theta^*) \geq \langle \nabla \mathcal{L}(\theta^*), \Delta \rangle + \frac{\kappa_l}{2} \|\Delta\|_2^2 .$$

We can establish

$$\begin{aligned} \langle \nabla \mathcal{L}(\theta^*), \Delta \rangle &\geq - |\langle \nabla \mathcal{L}(\theta^*), \Delta \rangle| \\ &\geq - \|\nabla \mathcal{L}(\theta^*)\|_\infty \|\Delta\|_1 = \lambda_n \|\Delta\|_1 , \end{aligned}$$

and hence,

$$\begin{aligned} G(\theta^* + \Delta) &\geq -\lambda_n \|\Delta\|_1 + \frac{\kappa_l}{2} \|\Delta\|_2^2 - \sqrt{2 \left| S^* - \widehat{S} \right| \kappa_u \epsilon_{\mathcal{S}}} \|\Delta\|_2 \\ &> \|\Delta\|_2 \left(\frac{\kappa_l}{2} \|\Delta\|_2 - \lambda_n \sqrt{\left| S^* \cup \widehat{S} \right|} - \sqrt{2 \left| S^* - \widehat{S} \right| \kappa_u \epsilon_{\mathcal{S}}} \right) \\ &> 0, \end{aligned}$$

if $\|\Delta\|_2 = r$ for

$$r = \frac{2}{\kappa_l} \left(\lambda_n \sqrt{\left| S^* \cup \widehat{S} \right|} + \sqrt{2 \left| S^* - \widehat{S} \right| \kappa_u \epsilon_{\mathcal{S}}} \right) .$$

This concludes the proof of the lemma.

□

Next, we note that when the inner while loop (forward step) returns, the backward step with the current parameters has failed to go through. This entails that

$$\inf_{j \in \widehat{S}} \mathcal{L}(\widehat{\theta} - \widehat{\theta}_j e_j) - \mathcal{L}(\widehat{\theta}) > \epsilon_{\mathcal{S}}/2. \tag{A.4}$$

The next lemma shows the consequence of this bound. **Lemma 3** (Stopping Backward Step). When the algorithm stops with parameter $\widehat{\theta}$ supported on $\widehat{S}$, we have

$$\left\|\widehat{\Delta}_{\widehat{S}-S^*}\right\|_2^2 \geq \frac{\epsilon_{\mathrm{S}}}{\kappa_u}\left|\widehat{S}-S^*\right|. \tag{A.5}$$

Proof. We have

$$\begin{aligned}
|\widehat{S}-S^*|\inf_{j\in\widehat{S}}\mathcal{L}(\widehat{\theta}-\widehat{\theta}_j e_j) &\leq \sum_{j\in\widehat{S}-S^*}\mathcal{L}(\widehat{\theta}-\widehat{\theta}_j e_j)\\
&\leq |\widehat{S}-S^*|\mathcal{L}(\widehat{\theta})+\sum_{j\in\widehat{S}-S^*}\left(\nabla_j\mathcal{L}(\widehat{\theta})\,\widehat{\theta}_j+\frac{\kappa_u}{2}\widehat{\theta}_j^2\right)\\
&\leq |\widehat{S}-S^*|\mathcal{L}(\widehat{\theta})+\frac{\kappa_u}{2}\left\|\widehat{\Delta}_{\widehat{S}-S^*}\right\|_2^2,
\end{aligned}$$

where the second inequality uses the fact that $[\nabla\mathcal{L}(\widehat{\theta})]_{\widehat{S}}=0$. Substituting (A.4) above, the lemma follows. □

Lemma 4 (Stopping Size). If $\epsilon_{\mathrm{S}} > \frac{\lambda_n^2}{\kappa_u}\left(\frac{\frac{1}{2\rho}\sqrt{\gamma}-\sqrt{\frac{\rho^2-\rho}{k^*}}}{\sqrt{1+\gamma}}-\sqrt{\frac{2}{2+\gamma}}\right)^{-2}$ and $RSC\left((2+\gamma)k^*\right)$ holds for some $\gamma \geq 4\rho^2\left(\sqrt{\frac{\rho^2-\rho}{k^*}}+\sqrt{2}\right)^2$, then the inner while loop (forward step) returns with $k \leq (1+\gamma)k^*$.

Proof. Consider the first time the inner while loop (forward step) reaches $k = (1+\gamma)k^*+1$, then by Lemma 3 and 2, we have

$$\begin{aligned}
\sqrt{\frac{k-1-k^*}{k-1}} \leq \sqrt{\frac{|\widehat{\mathcal{N}}_r^{(k-1)}-S^*|}{|\widehat{\mathcal{N}}_r^{(k-1)}\cup S^*|}} &\leq \frac{2\kappa_u\sqrt{\kappa_u(\kappa_u-\kappa_l)}}{\kappa_l^2\sqrt{|\widehat{\mathcal{N}}_r^{(k-1)}\cup S^*|}}+\frac{2\kappa_u}{\kappa_l}\left(\frac{\lambda_n}{\sqrt{\kappa_u\epsilon_{\mathrm{S}}}}+\sqrt{\frac{2|S^*-\widehat{\mathcal{N}}_r^{(k-1)}|}{|S^*\cup\widehat{\mathcal{N}}_r^{(k-1)}|}}\right)\\
&\leq \frac{2\frac{\kappa_u}{\kappa_l}\sqrt{\left(\frac{\kappa_u}{\kappa_l}\right)^2-\frac{\kappa_u}{\kappa_l}}}{\sqrt{k-1}}+\frac{2\kappa_u}{\kappa_l}\left(\frac{\lambda_n}{\sqrt{\kappa_u\epsilon_{\mathrm{S}}}}+\sqrt{\frac{2k^*}{k+k^*-1}}\right).
\end{aligned}$$

Hence, we get

$$\frac{\frac{1}{2\rho}\sqrt{\gamma} - \sqrt{\frac{\rho^2-\rho}{k^*}}}{\sqrt{1+\gamma}} - \sqrt{\frac{2}{2+\gamma}} \leq \frac{\lambda_n}{\sqrt{\kappa_u \epsilon_{\mathcal{S}}}}.$$

For $\gamma \geq 4\rho^2 \left(\sqrt{\frac{\rho^2-\rho}{k^*}} + \sqrt{2}\right)^2$, the LHS is positive and we arrive to a contradiction with the assumption on $\epsilon_{\mathcal{S}}$.

□

Bibliography

[1] A. Agarwal, S. Negahban, and M. Wainwright. Convergence rates of gradient methods for high-dimensional statistical recovery. In *Neural Information Processing Systems (NIPS)*, 2010.

[2] C. Aliferis, I. Tsamardinos, and L. Brown. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65:31–78.

[3] AllMusic. Allmusic.com, 2011.

[4] D. Chickering. Learning bayesian networks is np-complete. *Proceedings of AI and Statistics*, 1995.

[5] D. Chickering. Learning bayesian networks is np-complete. In *Proceedings of AI and Statistics*, 1995.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–421, 2008.

[7] N. Friedman and D. Peer I. Nachaman. Learning bayesian nework structure from massive datasets: The sparse candidate algorithm. In *Uncertainty in Artificial Intelligence (UAI)*, 1999.

[8] A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *The Pacific Symposium on Biocomputing (PSB)*, 2001.

[9] InfoChimps. Audioscrobbler dataset, May 2005.

[10] A. Jalali, C. Johnson, and P. Ravikumar. On learning discrete graphical models using greedy methods. In *Neural Information Processing Systems (NIPS) (currently under review)*, 2011.

[11] A. Jalali, P. Ravikumar, V. Vasuki, and S. Sanghavi. On learning discrete graphical models using group-sparse regularization. *AISTATS*, 14, 2011.

[12] D. Koller and N. Friedman. *Probabilistic Graphical Models*. MIT Press, Cambridge, MA, 2009.

[13] last.fm. last.fm, 2011.

[14] D. Margaritis. Learning bayesian networks model structure from data. *Ph.D. thesis, CMU*, 2003.

[15] N. Meinshausen and P. Buhlmann. High dimensional graphs and variable selection with the lasso. In *Annals of Statistics*, volume 34, pages 1436–1462, 2006.

[16] M. Mitzenmacher and E. Upfal. *Probability and Computing*. Cambridge University Press, New York, NY, 2009.

[17] S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Neural Information Processing Systems (NIPS)*, volume 22, 2009.

[18] J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, New York, NY, 1988.

[19] Pandora Radio. Pandora radio, 2011.

[20] P. Ravikumar, M. Wainwright, and J. Lafferty. High-dimensional ising model selection using ℓ_1-regularized logistic regression. *Annals of Statistics*, 38(3):1287–1319, 2010.

[21] A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic journal of statistics*, 2:494–515, 2008.

[22] P. Spirtes, C. Glymour, and R. Schneines. Causation, prediction and search. *MIT Press*, 2000.

[23] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

[24] T. Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Neural Information Processing Systems (NIPS)*, volume 21, 2008.

[25] T. Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568, 2009.

Vita

Christopher Carroll Johnson was born in Madison, CT. He received his Bachelor of Science in Computer Science with Honors from The University of Texas at Austin in May 2006. During this time he also completed an internship with the World Wide Web Consortium (W3C) at MIT where he worked with Dr. Ralph Swick and Dr. Push Singh on a semantic web gift recommendation engine. Upon completing his degree he spent a year as a software engineer for Lockheed Martin's Maritime Systems and Sensors in Manassas, VA where he designed software for undersea surveillance systems. After this he spent 2 years as a software engineer for Blackboard Inc. in Washington D.C. where he wrote learning and content management web application software. In February 2009 he travelled to Bangkok, Thailand where he taught high school Mathematics at Bangkok Christian College for the next 9 months. In January 2010 he began his graduate studies in the Department of Computer Sciences at The University of Texas at Austin. In the Summer of 2010 he completed an internship at MIT Lincoln Laboratory where he designed a supervised learning named entity recognizer for semi-structured data under the guidance of Dr. Michael Yee. In the Summer of 2011 he completed an internship with the LINQS group at The University of Maryland College Park where he investigated new methods of relational clustering using Probabilistic Soft Logic (PSL) under the guidance of Professor Lise Getoor. Christopher is currently working towards his PhD

in Computer Science at The University of Texas at Austin with a focus in Statistical Machine Learning under the supervision of his advisor Pradeep Ravikumar. This Document serves as partial completion of his Masters of Science in Computer Science degree.

Contact: CJohnson [at] cs [dot] utexas [dot] edu

This thesis was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.